

(~) tilde

encapsulation  $\Rightarrow$  إخفاء معلومات البرنامج واستخدامها في ملف خاص ويتم باستدائه ب (h) header

implementation (تنفيذ)

class 11/11/11 في بداية البرنامج

```
#ifndef _Date_h_  $\rightarrow$  if not define
#define _Date_h_
#include <iostream.h>
class Date {
    !
```

ملاحظة

Constructor

$\rightarrow$  default }  
 $\rightarrow$  overloading #endif في النهاية

# دالة 11 main (في ملف ثاني)

Source file

```
#include <iostream.h>
#include <Date.h>
int main()
{
    Date d1;
    }
}
```

كتابة Date 11



- Symbol

printf (format, variable)

↓  
قائمة الخرج  
توظيف الخرج

Q2 → Sheet # Struct

Functions : إيجاد الزوايا

→  $\tan^{-1}$ 

∴ 4 Fns.

ملاحظة :  $\tan^{-1}$ 

mult = multiplication (comp1, comp2); الزاوية

theta1 = atan (comp1.real / comp1.img) \* (180 / 3.14);

theta2 = atan (comp2.real / comp2.img) \* (180 / 3.14); عند

theta3 = atan (add.real / add.img) \* (180 / 3.14); أجب

theta4 = atan (sub.real / sub.img) \* (180 / 3.14); الشيء

theta5 = atan (mult.real / mult.img) \* (180 / 3.14);

↑  
img / real      ← الخرج

printf (format, variable)

ex.

→ decimal

# (أو متغير واحد)

printf ("Area = %d", area)

%f Float

%c character

# (أو أكثر من متغير الصحيح في 8 خانة) → %3f ← العشري في 3 خانة

printf ("%f length %f width %d Area", length, width, Area);  
%f %f %d





```

class Square
{
private:
    int side;
public:
    void setSide (int s);
    int getSide ();
};
// prototype
void main() {
int Square :: getSide ()
{
    return side;
}
}

```

$a=2, b=4$   
 base ()  
 (5, 2)

ex. class A {

```

    int x →
public:
    int get()
{
    return x;
}

```

لو مالتش private هتلازم تكتب default value  
لو مالتش private هتلازم تكتب default value

⇒ Constructors: بيأخذ نفس اسم ال class

# default Constructors

- لو مالتش أرقام بيده = 1
- كل المتغيرات مالتش قيم
- كل المتغيرات ليها قيم
- قيمه ابتدائية واحدة

Invoking a Constructor

constructor stamul

Square square1;

# Overloading

قد يكون ال class على أكثر من

Constructor

ex.

Square ();  
 Square (int);

← لازم ال parameters  
 تكون مالتش



class ← مجموعة من الobjects تشترك في نفس الخصائص

⊙ Attributes: int, string, ... etc

⊙ Methods:

1. Constructor & destructor (إنشاء وهدم)

2. Modifier (Set fn.)

3. Access (Get fn.)

4. Other function (like display)

Unified Modelling Languages (UML)

|            |
|------------|
| Class Name |
| Attributes |
| Methods    |

Flowchart الـ class

ex. <sup>Small letter</sup> class className ~ Name (int a)

}

declaration;

};

ex. class Square



{ private:

int side;

access  
specifiers

public:

void setSide (int s)

{ side = s; }

int getSide ()

{ return side; }

};

void main ()

{ Square sq1, sq2;

sq1.setSide(5);

cout << sq2.getSide();

};

|                 |
|-----------------|
| Square          |
| Side: int       |
| setSide(): void |
| getSide(): int  |

→ set side

→ get side

→ prototype method

class operation

→ outline fn.

inline  
Function  
class operation





## → Enumerators

نوع

الوظيفة : يجمع بعض المتغيرات التي يمكن حصرها

enum type {

value 1;

value 2;

}; Object . name;

ex.

enum color {

black, blue,

green, red,

purple, white

};

Color myColor;

myColor = red;

→ enum object

enum color {

red = 3, green<sup>4</sup>, blue<sup>5</sup>,white = 8, black<sup>9</sup>};

ex. enum Months {

January = 1, February, March, April,

};

خصصنا أول متغير بقيمة 1

هو ثانياً حسب 2, 3, الخ لأنه غير

بعض المتطلبات :

## → Object Oriented (oop) object oriented programming

• متغير في class وان هناك متغيرات لها (تقبل المعلومات) وصف

## → (Data encapsulation, information Hiding) class

نظم وإخفاء المعلومات لأشخاص غير المرغوب

## → Data Abstraction

الوقت ليس وصف المعلومات وتبسيط

المبادئ الأساسية لـ oop :

# Object # classes

# encapsulation # Inheritance

# poly.



```

cout << "sum=" << add.real << "+" << add.img << endl;
cout << "sub=" << sub.real << "+" << sub.img;
return 0;
}

```

ملاحظة 5:-

### (Unions & Enumerators)

Union → Struct أو  
تخزين قيم واحدة فقط

```

Union wageInfo
{
    double hourlyRate;
};

```

→ Anonymous Union

Variable tag في Union

Structure with regular union

```

Struct book {
    Char title[50];
    Char author[50];
    Union {
        Float dollars;
        int yen;
    } price;
};
book1;

```

Struct (tag) Union هي في الـ Union

→ book1.price.dollars

structure with anonymous Union

```

Struct book {
    Char title[50];
    Char author[50];
    Union {
        Float dollars;
        int yen;
    };
};
book2;

```

→ book2.dollars



Q2.

#include &lt;iostream&gt;

Using namespace std;

Struct Complex {

float real, img;

};

Complex addition (Complex c1, Complex c2) definition

{

Complex c;

c.real = c1.real + c2.real;

c.img = c1.img + c2.img;

return c;

}

Complex subtraction (Complex c1, Complex c2)

{ Complex c;

c.real = c1.real - c2.real;

c.img = c1.img - c2.img;

return c;

}

Void print (Complex c)

{ cout &lt;&lt; "(" &lt;&lt; c.real &lt;&lt; ", " &lt;&lt; c.img &lt;&lt; ")"; }

⇒

int main()

{ Complex c1, c2;

Complex add, sub;

cout &lt;&lt; "Enter first complex No.:";

cin &gt;&gt; c1.real &gt;&gt; c1.img;

cout &lt;&lt; "Enter second complex No.:";

cin &gt;&gt; c2.real &gt;&gt; c2.img;

add = addition (c1, c2);

sub = subtraction (c1, c2);

calling ←

print (c1);

print (c2);





Sheet #3

Q1.

نوع ال Struct خارج  
main ال

#include &lt;iostream&gt;

Using namespace std;

struct Rectangle

{ float width, length;

};

float Area(float w, float L)

definition

{

return w \* L;

(;) ~~المعروف~~

}

float per(float w, float L)

{

return 2 \* (w + L);

}

void print(float w, float L)

{

cout &lt;&lt; "(" &lt;&lt; w &lt;&lt; ", " &lt;&lt; L &lt;&lt; ")" &lt;&lt; endl; (5, 6)

→ int main()

(6, 7)

{

Rectangle r;

نوع ال struct المتغير  
cout << "Enter width and length of Rec :";

cin &gt;&gt; r.width &gt;&gt; r.length;

float area, perimeter;

area = Area(r.width, r.length);

perimeter = per(r.width, r.length);

print(r.width, r.length);

cout &lt;&lt; "The area = " &lt;&lt; area;

cout &lt;&lt; "The perimeter = " &lt;&lt; perimeter;

}



### ⇒ Members of Nested Structures:-

```
Student S5;  
S5.pData.name = "Ahmed";  
S5.pData.city = "Tanta";
```

### ⇒ Passing members of struct to functions.

```
ComputeGPA (S1.gpa);
```

### ⇒ Returning a structure from a function

```
(tag)  
Student getStuData(); ⇒ prototype  
S1(); ⇒ calling  
(Variable)
```

```
int x ( );
```



25

لو الأول والثالث: 12 و 6 و 5

Dimensions box1 = {12, 6, 5}; valid

Dimensions box2 = {12, 3, 5}; Invalid

default value لو الـ 3 is 1 #

## [2] Constructor

member جال داخل struct 1

inside the struct 2

Struct Dimensions

{ int length, width, height;

Dimensions (int L=1, int W=1, int H=1)

{ length=L; width=W; height=H; }

};

## → Struct of constructor

① Dimensions box3 (12, 3, 5);

② Dimensions box4 (12, 6);

③ Dimensions box5;

← default value 3 is 1

الـ 3 لو الـ 1

## → Nested Structures

Struct PersonInfo

{ string name, address, city;

};

Struct Student

{ int StudentID;

PersonInfo pData; name, address, city

Short year;

double gpa;

};





طرق الطباعة :-

cout << s1; ~~X~~ Invalid

→ cout &lt;&lt; s1.StudentID;

cout &lt;&lt; s1.name;

cout &lt;&lt; s1.year;

cout &lt;&lt; s1.gpa;



أحد المتغير التابع

s1

(بخصوص)

⇒ Comparing Struct members :-

if (s1 > s2) ~~X~~ Invalid

→ if (s1.gpa &gt; s2.gpa)

⇒ Initializing a Structure

وضع القيم الابتدائية

int StudentID = 1145; ~~X~~ InvalidString name = "Ali"; ~~X~~ Invalid

ممنوع أن نعط قيم ابتدائية على الـ Struct على هذا الشكل

1. initialization list

2. Constructor

↓ بطريقتين

[1] Initialization list:

Struct Dimensions

{

int length, width, height;

};

Dimensions box = {12, 6, 3};

length

height

↓  
widthtag of the  
structure↓  
s1  
الاسم  
للـ  
struct

محاضرة 4 :- م. . . . .

⇒ Structure :- is a c++ construct that allows multiple variables to be grouped together.

# يسمح بتخزين أكثر من

قيمة في الذاكرة

Struct Structure\_name

↳ tag: capital أو يكتوب

Structure member

```

{
    type1 field_1;
    type2 field_2;
    ...
    typen field_n;
}

```

Student s1; ⇒ جوة ال main

or

```

struct Student
{

```

int studentID;

String name;

Short year;

double gpa;

}; s1; variable

⇒ Accessing Structure members:

# طرق مختلفة لإدخال القيم :-

cout << "Enter the Student name:";

① getline ( cin, s1.name );

cout << "Enter the Student ID:";

② cin >> s1.studentID;

③ s1.gpa = 3.75;





// Function definition

```
void dbl_By_Address (int *a_ptr)
```

```
{ *a_ptr = 2 * (*a_ptr);
```

```
}
```

⇒ output

i before passing its pointer to the function = 4

i after execution of the function = 8

\* Swapping variable values :-

تبادل قيم المتغيرات

✓ 3 مثال

```
#include <iostream.h>
```

```
void swapInt (int *, int *);
```

```
int main ()
```

```
{ int i=35, j=50;
```

```
cout << "In Before swapping : i = " << i << " j = " << j;
```

```
swapInt (&i, &j);
```

```
cout << "In After swapping : i = " << i << " j = " << j;
```

```
cout << endl;
```

```
return 0;
```

```
}
```

```
void swapInt (int *a_ptr, int *b_ptr)
```

```
{ int temp;
```

```
temp = *a_ptr;
```

```
*a_ptr = *b_ptr;
```

```
*b_ptr = temp;
```

```
}
```

⇒ Output :-

Before swapping : i = 35, j = 50

After swapping : i = 50, j = 35





cat

cout << endl;  
return 0;

⇒ Output :-

Enter the name of the salesperson : Ahmed Ibrahim

Enter the sales for Monday : 650.6

" " " " Tuesday : 438.9

" " " " Wednesday : 321.5

" " " " Thursday : 750.8

" " " " Friday : 862.2

" " " " Saturday : 651.0

" " " " Sunday : 411.8

The total sales for Ahmed Ibrahim are 4086.80.

The highest sales were 862.20.

The highest sales occurred on Friday.

### Pointers & Arrays & Functions

المسائل التي تتطلب قيمة عن طريق الـ address

#include <iostream.h>

void db1 - By - Address (int x);

int main()

{ int i = 4;

cout << "In i before <sup>its</sup> passing pointer to the function." << i;

db1 - By - Address (&i);

cout << "In i after the execution of the function=" << i;

cout << endl;

return 0;

2 زوایر



## Marketing

# اكتب برنامج لحساب مبيعات مندوبي اقسام المبيعات والبيع الذي  
صنع ذلك

```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
Char dayNames [7][10] = {"Monday", "Tuesday",  
"Wednesday", "Thursday", "Friday", "Saturday",  
"Sunday"};
```

```
double sales [7]; → "10" لو جود 8 لو جود 10  
Char salesperson [41]; [40+1] String او
```

```
int day, maxDay;
```

```
cout << "In Enter the name of the salesperson :";
```

```
cin.getline (salesperson, 41);
```

```
for ( day = 0; day <= 7; ++day)
```

```
{ cout << "In Enter the sales for" << dayNames [day] << " :";
```

```
cin >> sales >> sales [day];
```

```
}
```

```
total sales = 0; maxDay = 0; max sales = sales [0];
```

```
for ( day = 0; day <= 7; ++day) ++day لا تكتب
```

```
{ if ( sales [day] > max sales) day++ لا تكتب
```

```
{ max sales = sales [day];
```

```
maxDay = day;
```

```
}
```

```
total sales += sales [day];
```

```
}
```

```
cout << "In The total sales for" << sales person <<  
"are" << total sales << " .";
```

```
cout << "In The highest sales were" << max sales <<
```

```
cout << "In The highest sales occurred on" <<  
dayNames [maxDay] << " .";
```





ملاحظة 3 :-

# المصفوفة المكونة من سلاسل حرفية

Array Consists of strings of Characters:-

Char cityNames[6][9] = { "Cairo", "Amman", "NewYork", "London", "Beirut", "Khartoom" }; 20

|   |   |   |   |   |    |    |   |    |
|---|---|---|---|---|----|----|---|----|
| C | a | i | r | o | \0 |    |   |    |
| A | m | m | a | n | \0 |    |   |    |
| N | e | w |   | y | o  | r  | k | \0 |
| L | o | n | d | o | n  | \0 |   |    |
| B | e | i | r | u | t  | \0 |   |    |
| K | h | a | r | t | o  | o  | m | \0 |

1D لو

|    |
|----|
| C  |
| a  |
| i  |
| r  |
| o  |
| \0 |
| 1  |
| 2  |
| 3  |
| A  |
| m  |
| m  |







```
cin >> A[i][j];
```

```
}
```

```
cout << "\n Output matrix array";
```

```
for (i=0; i<3; i++)
```

المصفوفة

```
{ cout << "\n";
```

```
for (j=0; j<4; j++)
```

الصفحة

```
cout << " " << A[i][j];
```

```
}
```

```
return 0;
```

```
}
```

→ Output

Enter element of array

A[0][0] = 3

A[2][0] = 3

A[0][1] = 4

A[2][1] = 2

A[0][2] = 5

A[2][2] = 8

A[0][3] = 6

A[2][3] = 0

A[1][0] = 0

Output matrix array

A[1][1] = 2

3 4 5 6

A[1][2] = 4

0 2 4 5

A[1][3] = 5

3 2 8 0

← يمكن دمج أو تمثيل كل عنصر في 2 خلية

→ مصفوفة → Mat[i][j] = 2 \* A[i][j];

يتم تعريفها في بداية

البرنامج



= 2 خطا

# Write a program to enter an array with (2D) Using Keyboard and print it as follows:-

|   | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 3 | 4 | 5 | 6 |
| 1 | 0 | 2 | 4 | 5 |
| 2 | 3 | 2 | 8 | 0 |

### ⇒ Algorithm

\* Start

\* loop 1:  $i=0$  to  $n$  step 1

input  $A[i][j]$  → loop  $j=0$  to  $m$  step 1

End loop 1

input  $A[i][j]$

\* loop 2  $i=0$  to  $n$  step 1

end loop  $j$

loop 3  $j=0$  to  $m$  step 1

end loop  $i$

Output  $A[i][j]$

End loop 3

End loop 2

\* End

### ⇒ The program

/\*

\*/

#include <iostream>

int main()

{ int A[3][4];

int i, j;

cout << "Enter elements of array \n";

for ( $i=0$ ;  $i<3$ ;  $i++$ ) الصفوف

{ for ( $j=0$ ;  $j<4$ ;  $j++$ ) العمود

cout << "A[" << i << "]" << j << "]= ";





$$= 1 \text{ expts}$$

1 Byte = 8 Bits

## # Arrays :-

for(---) X  
cin >> x

$$\text{int ID}[5] = \{2, -3, 5, -15, 99\};$$
or
$$10[4] = 99;$$

```
int temp [2][4] = { { 2, -3, 5, -15 }, { 1, 2, 4, -10 } };
```

```
temp[0][0] = 2;
```

```
temp[1][3] = -10;
```

